



# EECS 270 Fall 2022

## Introduction to Logic Design

3. Timing and Delay

# Recap++

## □ Quiz 1, Question 1:

How many different numbers can be represented with 16 bits?

$$\boxed{\text{Max value} = 2^n - 1} \quad \xrightarrow{\text{number of bits}} \quad \begin{array}{l} \text{Max value} = 65,535 \\ \text{Min value} = 0 \end{array} \quad } \quad \left. \right\} \text{65,536 integer values in total}$$

$$\boxed{\text{Total number of integers} = 2^n}$$

# Recap++

## □ Quiz 1, Question 2:

How many bits are needed to represent the following values in binary?

a)  $10_8 = 1 * 8^1 + 0 * 8^0 = 8_{10}$   $\rightarrow \text{ceil}(\log_2(8+1)) = \text{ceil}(3.17) = 4$

b)  $10_{10}$   $\rightarrow \text{ceil}(\log_2(10+1)) = 4$

c)  $10_{16} = 1 * 16^1 + 0 * 16^0 = 16_{10}$   $\rightarrow \text{ceil}(\log_2(16+1)) = 5$

$$n = \text{ceil}(\log_2(\text{value}+1))$$

# Recap++

## □ Quiz 1, Question 3:

Perform the following binary-to-decimal and decimal-to-binary number conversions:

a)  $100001110_2 = 2^8 + 2^3 + 2^2 + 2^1 = 270_{10}$

b)  $2022_{10} = 11111100110_2$

c)  $100000_2 = 2^5 = 32_{10}$

d)  $212_{10} = 11010100_2$

e)  $111111111_2 = 2^8 + 2^7 + \dots + 2^1 + 2^0 = 511_{10}$   
 $= 2^9 - 1$

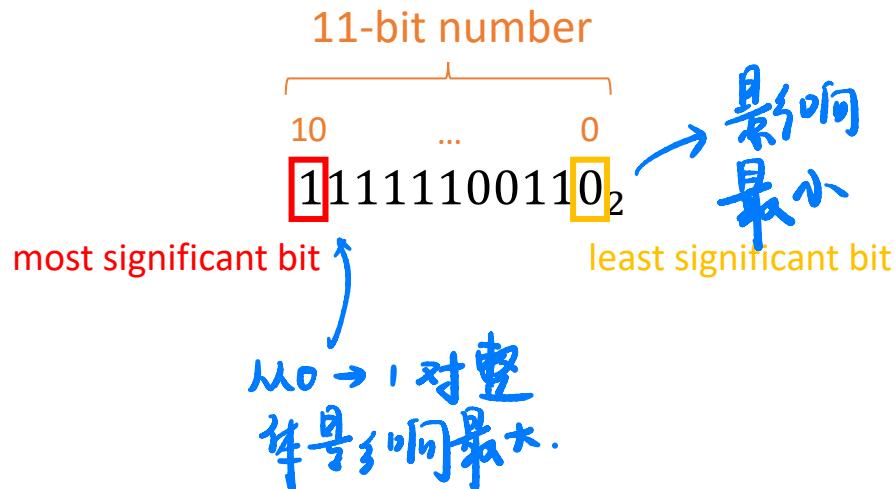
### Decimal-to-binary conversion:

- Step 1: Divide the given number repeatedly by 2 until you get 0 as the quotient.
- Step 2: Write the remainders in reverse order.

Step 1	Quotient	Reminder
$212/2$	106	0
$106/2$	53	0
$53/2$	26	1
$26/2$	13	0
$13/2$	6	1
$6/2$	3	0
$3/2$	1	1
$1/2$	0	1

Step 2:  $11010100$

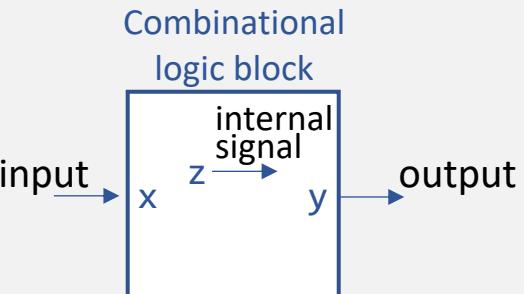
# Recap++



## Verilog

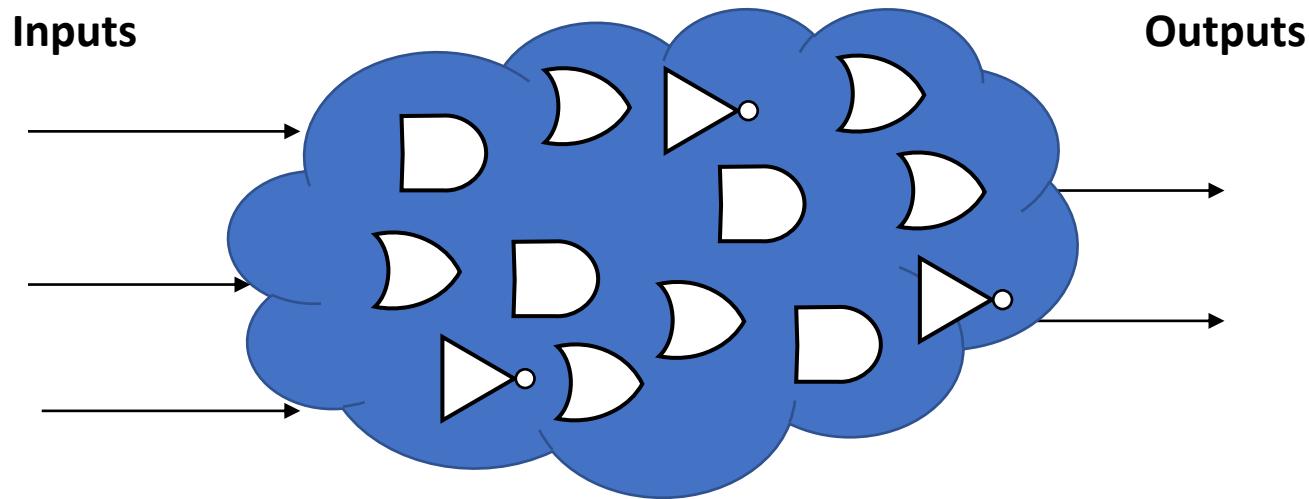
11 bits information

```
input [10:0] x;  
          most significant number found in position 10.  
output [10:0] y; least — in position 0.  
wire [10:0] z;
```



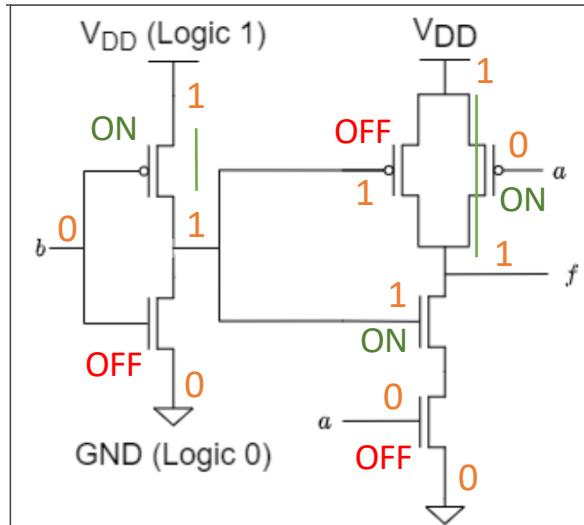
# Combinational logic

- ❑ Aka time-independent logic.
- ❑ Output is a pure function of the present input only.
- ❑ No feedback loops.



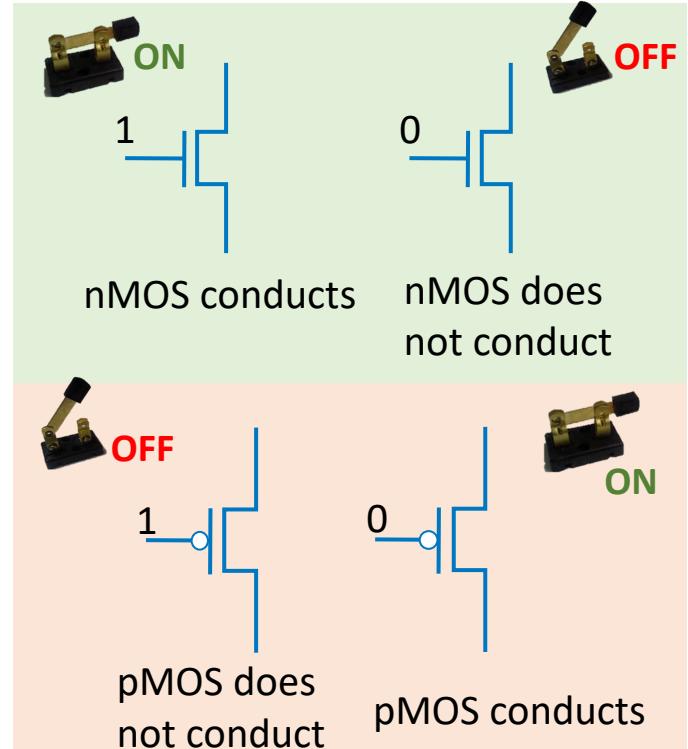
# Recap++

## □ Quiz 1, Question 4:



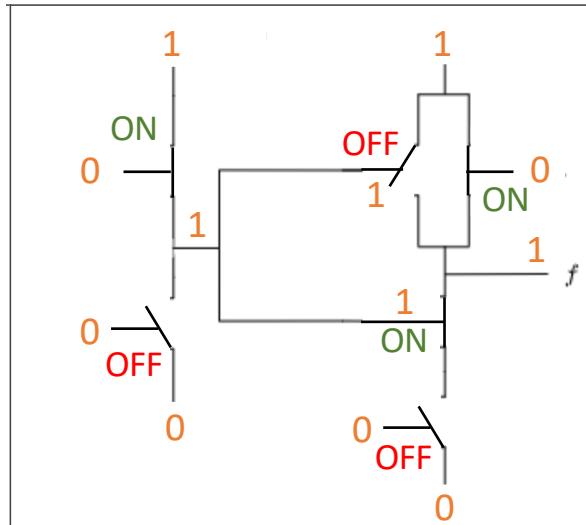
Truth table

a	b	f
0	0	1
0	1	
1	0	
1	1	



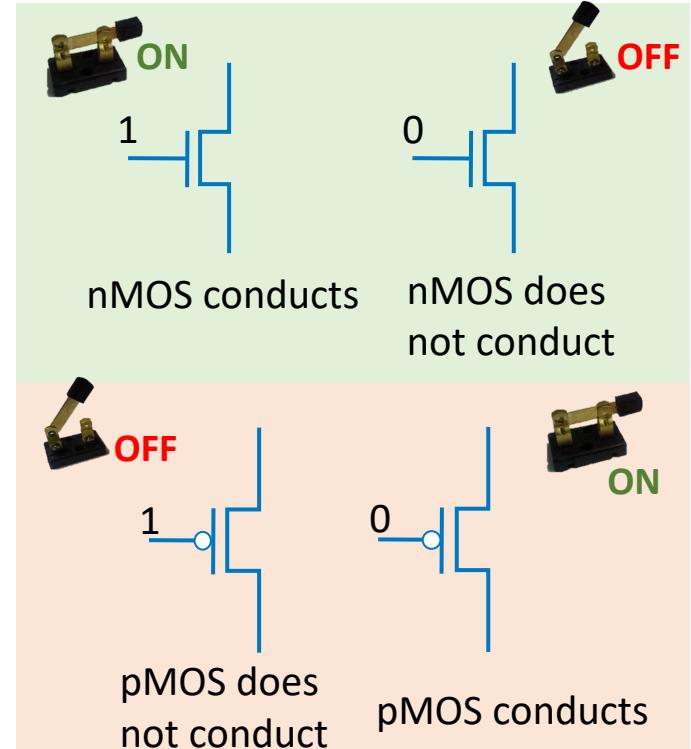
# Recap++

## □ Quiz 1, Question 4:



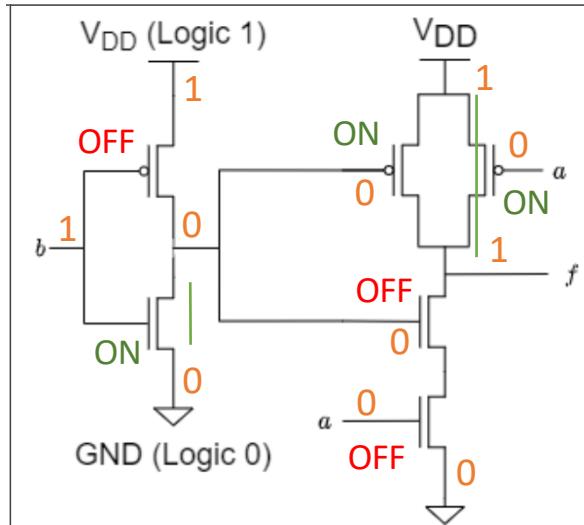
Truth table

a	b	f
0	0	1
0	1	
1	0	
1	1	



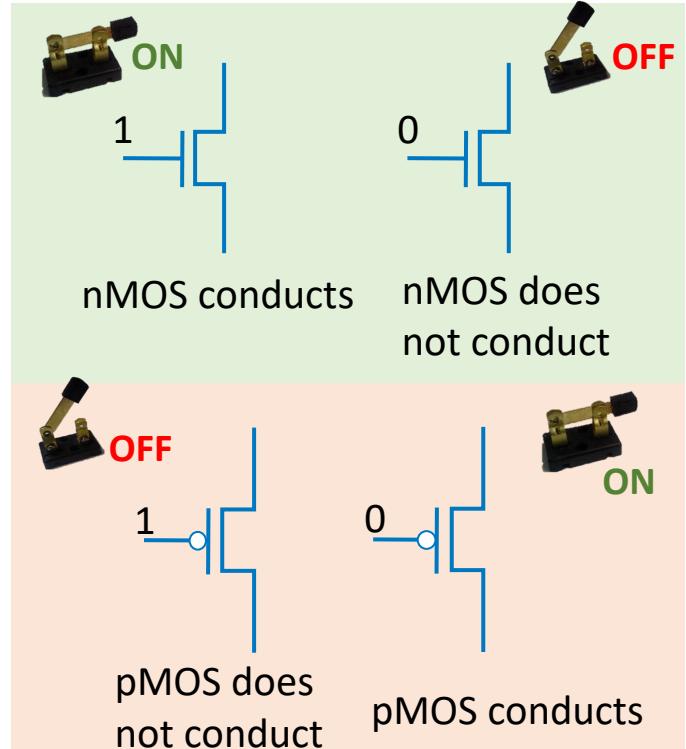
# Recap++

## □ Quiz 1, Question 4:



Truth table

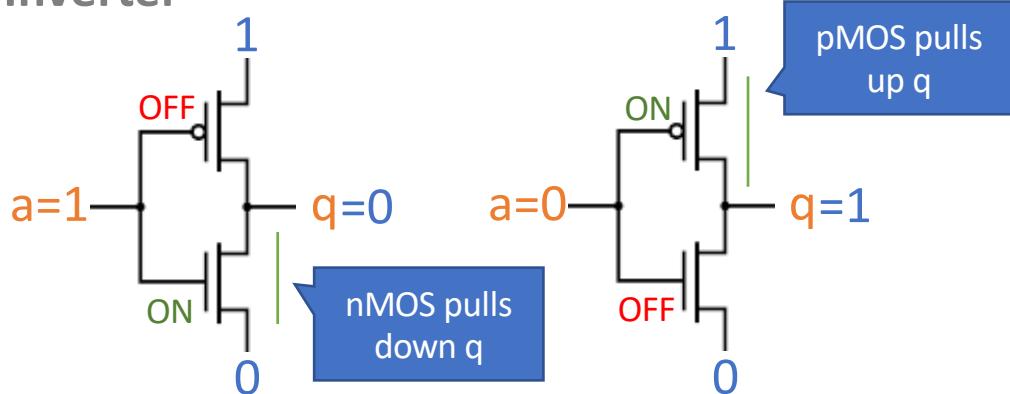
a	b	f
0	0	1
0	1	1
1	0	0
1	1	1



# Recap++

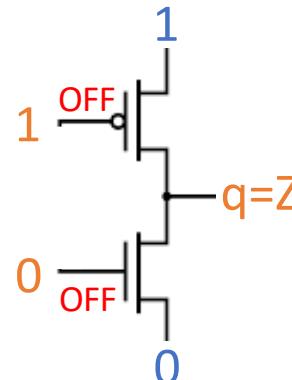
- Why use both pMOS and nMOS? In some cases, it seems redundant.
  - Transistors are not ideal switches
  - pMOS are good at propagating 1s
  - nMOS are good at propagating 0s

Inverter

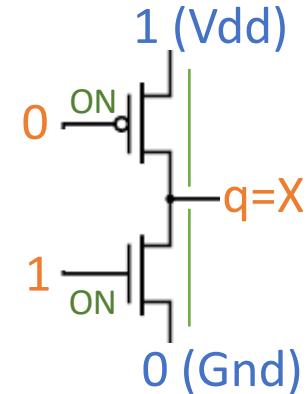


"illegal" states

High impedance



Short circuit



# Reminder

**This course is about 0s and 1s  
and their manipulation.**

**It is not about transistors.**

# Recap++

## □ Quiz 1, Question 5:

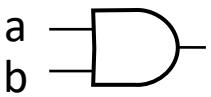
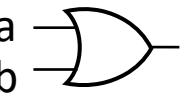
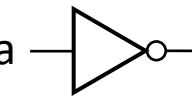
Four students are taking EECS 270. Two or three out of the four, but not all four, attended the review session for Exam 1. Let  $a, b, c, d$  denote Boolean variables that represent class attendance (1 for "present", 0 for "absent") of the four students. Which one of the following expressions precisely captures the stated attendance condition? In other words, which of the following Boolean expressions evaluates to true for the given data?

Note: Addition implies OR, multiplication implies AND, and apostrophe implies NOT

**Assumption:**  $a = 1, b = 1, c = 0, d = 0$

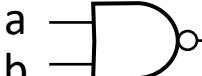
- $a'bc + ab'd + ac'd + bcd' = 0 \cdot 1 \cdot 0 + 1 \cdot 1 \cdot 0 + 1 \cdot 1 \cdot 0 + 1 \cdot 0 \cdot 1 = 0 + 0 + 0 + 0 = 0$
- $abcd' + abc'd + ab'cd + a'bcd = 1 \cdot 1 \cdot 0 \cdot 0 + 1 \cdot 1 \cdot 1 \cdot 0 + 1 \cdot 0 \cdot 0 \cdot 0 + 0 \cdot 1 \cdot 0 \cdot 0 = 0 + 0 + 0 + 0 = 0$
- $ab + ac + ad + bc + bd + cd = 1 \cdot 1 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 = 1 + 0 + 0 + 0 + 0 = 1$
- $(ab + ac + ad + bc + bd + cd)(a' + b' + c' + d') = 1 \cdot (0 + 0 + 1 + 1) = 1 \cdot 1 = 1$
- $abc + abd + acd + bcd = 1 \cdot 1 \cdot 0 + 1 \cdot 1 \cdot 0 + 1 \cdot 0 \cdot 0 + 1 \cdot 0 \cdot 0 = 0 + 0 + 0 + 0 = 0$

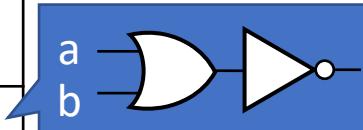
# Basic Boolean gates

	AND	OR	NOT
Schematic symbol			
Logic symbol	$a \cdot b$	$a+b$	$a'$ or $\bar{a}$
Verilog symbols	<code>&amp;&amp; or &amp;</code>	<code>   or  </code>	<code>! or ~</code>

Inputs	Outputs			
a	b	$a \cdot b$	$a+b$	$\bar{a}$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

# More Boolean gates

	NAND	NOR
Schematic symbol		
Logic symbol	$\overline{a \cdot b}$	$\overline{a+b}$
Verilog expression	$\sim(a \& b)$	$\sim(a   b)$

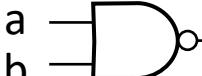


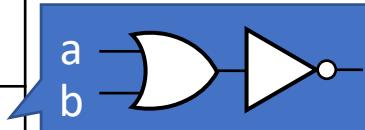
		Inputs	Outputs	
a	b	$\overline{a \cdot b}$	$\overline{a+b}$	
0	0		1	
0	1		0	
1	0		0	
1	1		0	

**In-class exercise:**

Fill out the shown truth table.

# More Boolean gates

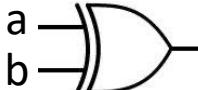
	NAND	NOR
Schematic symbol		
Logic symbol	$\overline{a \cdot b}$	$\overline{a+b}$
Verilog expression	$\sim(a \& b)$	$\sim(a   b)$



		Inputs	Outputs	
a	b	$\overline{a \cdot b}$	$\overline{a+b}$	
0	0	1	1	
0	1	1	0	
1	0	1	0	
1	1	0	0	

- NAND returns 1 when AND return 0, and vice versa
- NOR returns 1 when OR returns 0, and vice versa

# More Boolean gates

	XOR	XNOR
Schematic symbol		
Logic symbol	$a \oplus b$	$\overline{a \oplus b}$
Verilog expression	$a^b$	$\sim(a^b)$

is "1" only if the number of  
high-level input is odd

"1" even

"1" odd

"1" odd

1 even

		Inputs	Outputs	
a	b	$a \oplus b$	$\overline{a \oplus b}$	
0	0	0	1	
0	1	1	0	
1	0	1	0	
1	1	0	1	

## In-class exercise:

Write the Boolean functions that describe XOR and XNOR operations using only the basic AND, OR, NOT operators.

# More Boolean gates

$$\text{XOR}(a, b) = (\bar{a} \cdot b) + (a \cdot \bar{b})$$

## In-class exercise:

Write the Boolean functions that describe XOR and XNOR operations using only the basic AND, OR, NOT operators.

Inputs		Outputs	
a	b	$\overline{a \oplus b}$	$\overline{a \oplus b}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

# More Boolean gates

$$\text{XOR}(a, b) = (\bar{a} \cdot b) + (a \cdot \bar{b})$$

$$\text{XNOR}(a, b) = (\bar{a} \cdot \bar{b}) + (a \cdot b)$$

## In-class exercise:

Write the Boolean functions that describe XOR and XNOR operations using only the basic AND, OR, NOT operators.

Inputs		Outputs	
a	b	$\overline{a \oplus b}$	$a \oplus \overline{b}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

# Where in Canvas?

The screenshot shows the University of Michigan Canvas interface. On the left is a vertical navigation bar with various links:

- Fall 2022
- Home
- Announcements
- Piazza** (highlighted with a green box)
- Zoom
- Grades
- Assignments
- Gradescope
- Quizzes
- Discussions
- People
- Collaborations
- Engineering Honor Code
- Teaching Evaluations
- Files** (highlighted with a pink box)
- Engin Student Support

The main content area shows the "EECS 270 001 FA 2022 > Files" page. It includes a search bar, a breadcrumb trail, and a table of files. A large pink arrow points from the "Files" link in the sidebar to the "9. Additional Material" folder in the file list. A green arrow points from the "Piazza" link in the sidebar to the breadcrumb trail.

Name	Date Created	Date Modified	Modified By	Size
0. Euclid Font	Aug 22, 2021			--
1. Lectures	May 19, 2020			--
2. Labs	Jun 20, 2021			--
<b>9. Additional Material</b>	Yesterday			--

All My Files

# Additional resources

umich.instructure.com/courses/549093/pages/session-2-bits-everywhere

EECS 270 001 FA 2022 > Pages > Session 2: Bits Everywhere!

Fall 2022

View All Pages

Published

Session 2: Bits Everywhere!

Topics

- The digital abstraction
- The benefits of digitization
- Electronic switches: transistors
- Logic "gates"
- Combinational circuits

Suggested Reading

- Vahid: pp. 1-13, 35-45, 73-82

Lecture Files

- [Lecture2-Bits Everywhere!.pptx](#)
- [Lecture2-Bits Everywhere!.pdf](#)

Older Lecture Video by Instructors

- [Lecture2-WN2022](#)

- Preview before lecture
- Updated after lecture

# More updates and clarifications

	Monday	Tuesday	Wednesday	
9:00am				<a href="#">Link to google form</a>
10:00am			MCC w/ George @4777 BBB	
11:00am	Lecture @220 CHRYS	OH w/ Harrison virtual	Lecture @220 CHRYS	Revisit questions from last lecture
12:00pm				New material
30 minutes later				Practice exercises
1:00pm				
2:00pm	OH w/ George @4777 BBB		OH w/ Owen @BBB lobby	Quizzes:
3:00pm				<ul style="list-style-type: none"><li>- Open for 24 hours after lecture</li><li>- 3 tries per quiz—only best score counts</li><li>- 5 lowest scores don't count</li></ul>
4:00pm				

\* [Piazza site](#) also available for questions

***“In theory, theory and practice are the same. In practice, they are not.”***

Albert Einstein

## □ Theory

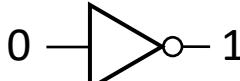
AND



OR

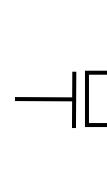


NOT

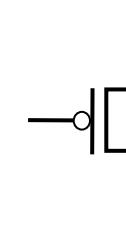


## □ Practice

nMOS



pMOS



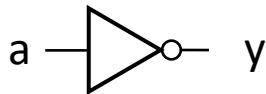
nMOS and pMOS do not propagate 0s and 1s equally well.

\*non-ideal

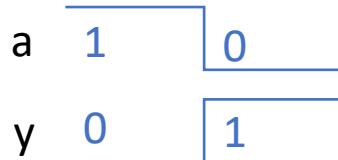


ON/OFF switching does not happen instantly

# Gate delays

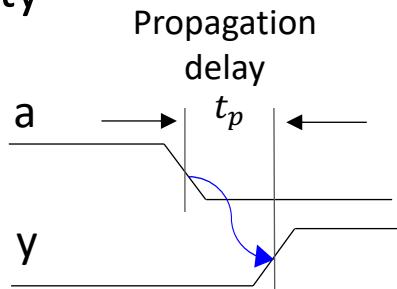


## ❑ Digital logic abstraction



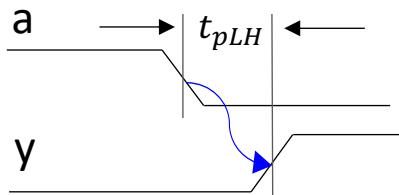
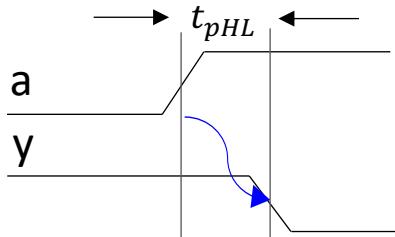
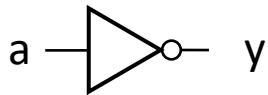
\*Output changes immediately with the input

## ❑ Reality



\*Output are delayed from inputs

# Gate delays



- $t_{pHL}$ : propagation delay from input to output when output is changing from high to low
- $t_{pLH}$ : propagation delay from input to output when output is changing from low to high

# Gate delays

- ❑ Delay in circuits is primarily caused by:
  - Resistance
  - Capacitance
  - Topology
  - Finite speed of light
- ❑ Anything affecting these quantities can change delay.
  - Rising ( $0 \rightarrow 1$ ) vs falling ( $1 \rightarrow 0$ ) inputs
  - Circuit aging
  - Changes in environment
    - circuits slow down when hot and speed up when cold

Interesting research area: superconductor electronics

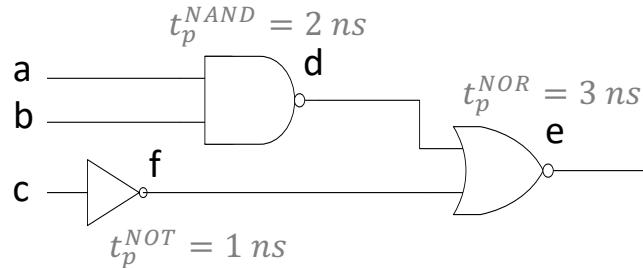
# Calculating delays across circuit paths

## Assumptions

- falling( $t_{pLH}$ ) and rising ( $t_{pLH}$ ) gate delays are the same
- $t_p^{NAND} = 2 \text{ ns}$ ,  $t_p^{NOR} = 3 \text{ ns}$ ,  $t_p^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task 1: Find the shortest delay path

Task 2: Find the longest delay path



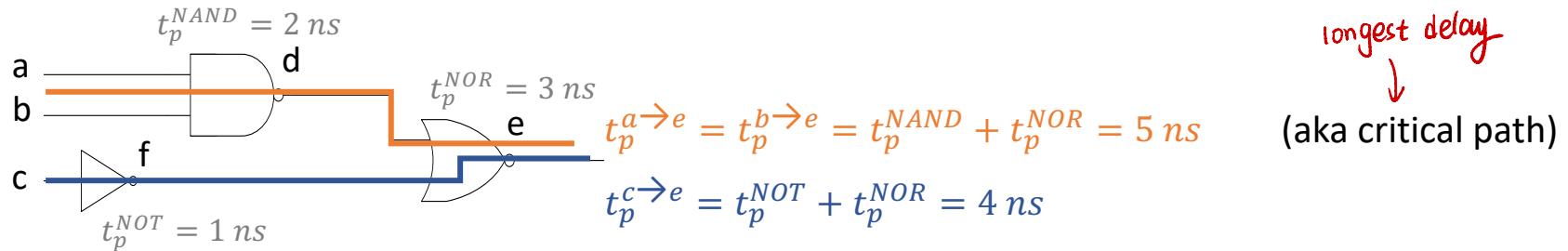
# Calculating delays across circuit paths

## Assumptions

- falling( $t_{pLH}$ ) and rising ( $t_{pLH}$ ) gate delays are the same
- $t_p^{NAND} = 2 \text{ ns}$ ,  $t_p^{NOR} = 3 \text{ ns}$ ,  $t_p^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task 1: Find the shortest delay path

Task 2: Find the longest delay path

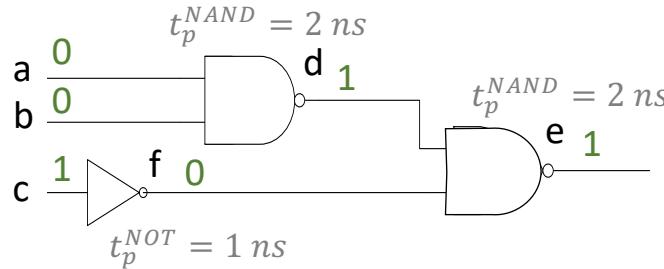


# Calculating delays across circuit paths

## Assumptions

- falling( $t_{pLH}$ ) and rising ( $t_{pLH}$ ) gate delays are the same
- $t_p^{NAND} = 2 \text{ ns}$ ,  $t_p^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task 1: Find the output of circuit for the provided inputs



Inputs		Outputs	
a	b	$\bar{a}$	$\bar{a} \cdot \bar{b}$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	0

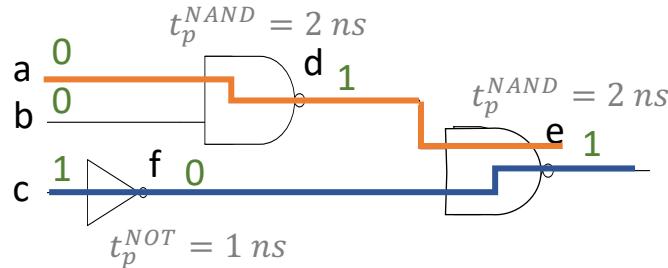
# Calculating delays across circuit paths

## Assumptions

- falling( $t_{pLH}$ ) and rising ( $t_{pLH}$ ) gate delays are the same
- $t_p^{NAND} = 2 \text{ ns}$ ,  $t_p^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task 1: Find the output of circuit for the provided inputs

Task 2: Find the longest/critical path and calculate its delay



$$t_p^{a \rightarrow e} = t_p^{b \rightarrow e} = t_p^{NAND} + t_p^{NAND} = 4 \text{ ns}$$
$$t_p^{c \rightarrow e} = t_p^{NOT} + t_p^{NAND} = 3 \text{ ns}$$

critical path

Inputs		Outputs	
a	b	$\bar{a}$	$\bar{a} \cdot \bar{b}$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	0

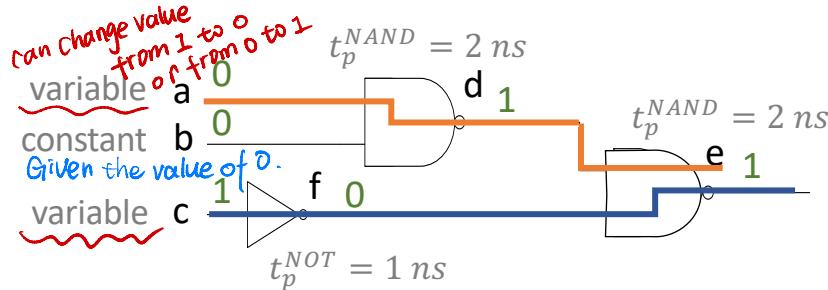
# Calculating delays across circuit paths

## Assumptions

- falling( $t_{pLH}$ ) and rising ( $t_{pLH}$ ) gate delays are the same
- $t_p^{NAND} = 2 \text{ ns}$ ,  $t_p^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task 1: Find the output of circuit for the provided inputs

Task 2: Find the longest/critical path and calculate its delay



Inputs		Outputs	
$a$	$b$	$\bar{a}$	$\bar{a} \cdot b$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	0

$$t_p^{a \rightarrow e} = t_p^{b \rightarrow e} = t_p^{NAND} + t_p^{NAND} = 4 \text{ ns}$$

$$t_p^{c \rightarrow e} = t_p^{NOT} + t_p^{NAND} = 3 \text{ ns}$$

critical path

Is the critical path still the same?

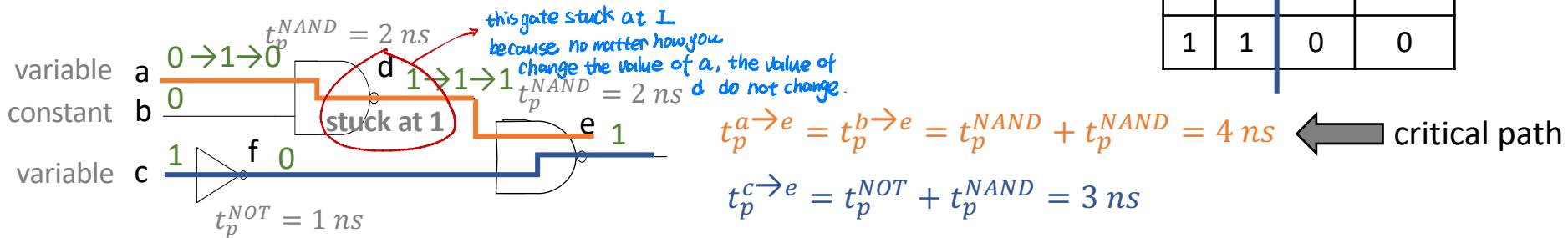
# Calculating delays across circuit paths

## Assumptions

- falling( $t_{pLH}$ ) and rising ( $t_{pLH}$ ) gate delays are the same
- $t_p^{NAND} = 2 \text{ ns}$ ,  $t_p^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task 1: Find the output of circuit for the provided inputs

Task 2: Find the longest/critical path and calculate its delay



Is the critical path still the same?

Inputs		Outputs	
a	b	$\bar{a}$	$\bar{a} \cdot \bar{b}$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	0

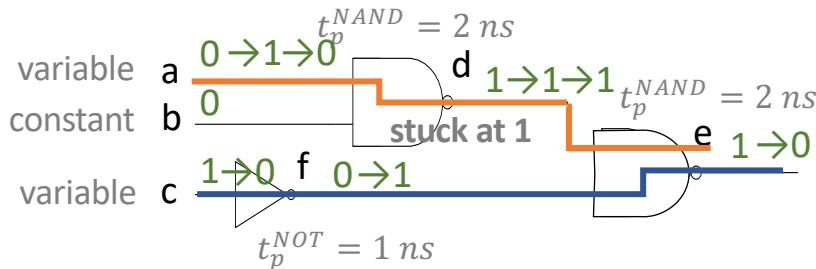
# Calculating delays across circuit paths

## Assumptions

- falling( $t_{pLH}$ ) and rising ( $t_{pLH}$ ) gate delays are the same
- $t_p^{NAND} = 2 \text{ ns}$ ,  $t_p^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task 1: Find the output of circuit for the provided inputs

Task 2: Find the longest/critical path and calculate its delay



$$t_p^{a \rightarrow e} = t_p^{b \rightarrow e} = t_p^{NAND} + t_p^{NAND} = 4 \text{ ns}$$

$$t_p^{c \rightarrow e} = t_p^{NOT} + t_p^{NAND} = 3 \text{ ns}$$

critical path

Is the critical path still the same?

		Inputs	Outputs		
		a	b	$\bar{a}$	$\bar{a} \cdot \bar{b}$
0	0	1	1	1	1
0	1	1	1	1	1
1	0	0	1	1	1
1	1	0	0	0	0

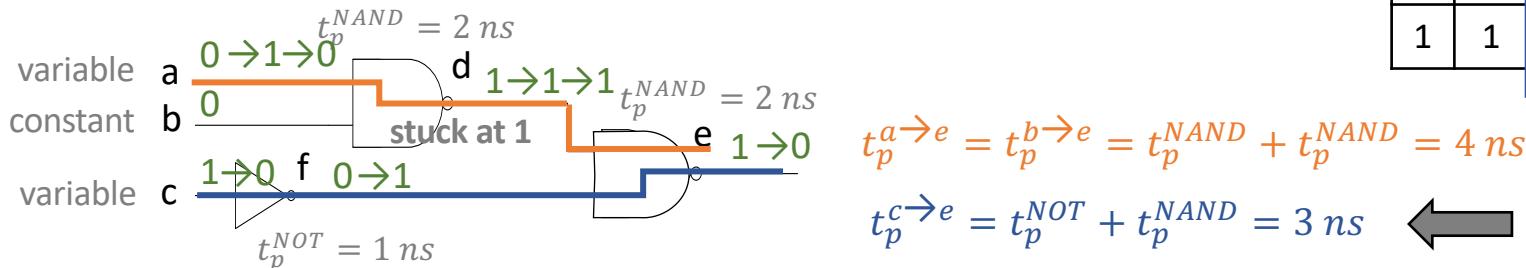
# Calculating delays across circuit paths

## Assumptions

- falling( $t_{pLH}$ ) and rising ( $t_{pLH}$ ) gate delays are the same
- $t_p^{NAND} = 2 \text{ ns}$ ,  $t_p^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task 1: Find the output of circuit for the provided inputs

Task 2: Find the longest/critical path and calculate its delay



Inputs		Outputs	
a	b	$\bar{a}$	$\bar{a} \cdot \bar{b}$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	0

The overall circuit delay depends on the transition path taken from the inputs to the output.

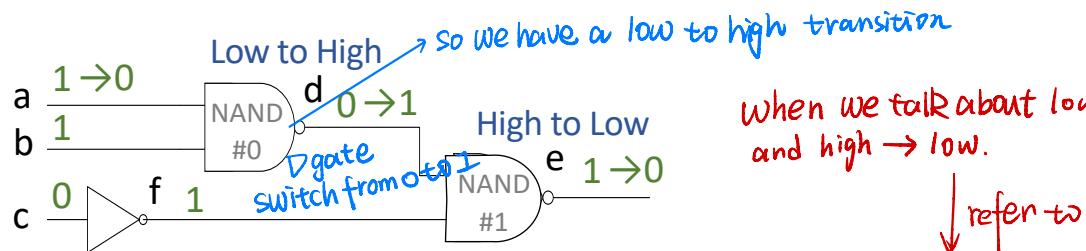
**Research:** Hardware verification, hardware security (timing side channels, etc.)

# Calculating delays across circuit paths

Assumptions

- $t_{pLH}^{NAND} = 4 \text{ ns}$ ,  $t_{pHL}^{NAND} = 3 \text{ ns}$ ,  $t_{pLH}^{NOT} = 2 \text{ ns}$ ,  $t_{pHL}^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task: Calculate  $t_p^{a \rightarrow e}$  when  $a$  goes from 1 to 0, while the other input signals remain unchanged



$$t_p^{a \rightarrow e} = t_{pLH}^{NAND\#0} + t_{pHL}^{NAND\#1} = 4 \text{ ns} + 3 \text{ ns} = 7 \text{ ns}$$

the outputs of the gate  
(but not the input of the gate)

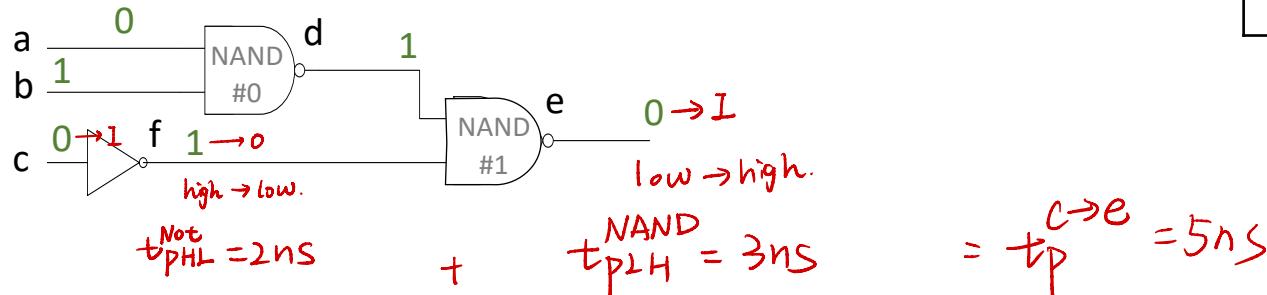
Inputs		Outputs	
a	b	$\bar{a}$	$\bar{a} \cdot \bar{b}$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	0

# Calculating delays across circuit paths

Assumptions

- $t_{pLH}^{NAND} = 4 \text{ ns}, t_{pHL}^{NAND} = 3 \text{ ns}, t_{pLH}^{NOT} = 2 \text{ ns}, t_{pHL}^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task: Calculate  $t_p^{c \rightarrow e}$ , when c goes from 0 to 1, while the other input signals remain unchanged



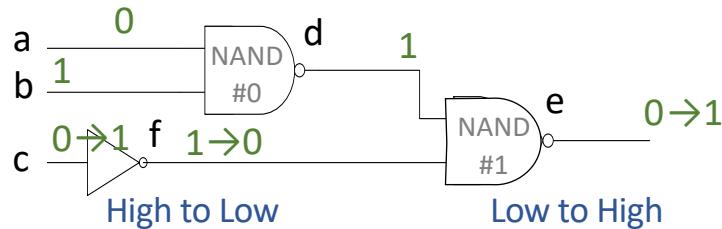
Inputs		Outputs	
a	b	$\bar{a}$	$\bar{a} \cdot \bar{b}$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	0

# Calculating delays across circuit paths

## Assumptions

- $t_{pLH}^{NAND} = 4 \text{ ns}, t_{pHL}^{NAND} = 3 \text{ ns}, t_{pLH}^{NOT} = 2 \text{ ns}, t_{pHL}^{NOT} = 1 \text{ ns}$
- Wires have zero delay

Task: Calculate  $t_p^{c \rightarrow e}$ , when c goes from 0 to 1, while the other input signals remain unchanged



$$t_p^{c \rightarrow e} = t_{pHL}^{NOT} + t_{pLH}^{NAND\#1} = 1 \text{ ns} + 4 \text{ ns} = 5 \text{ ns}$$

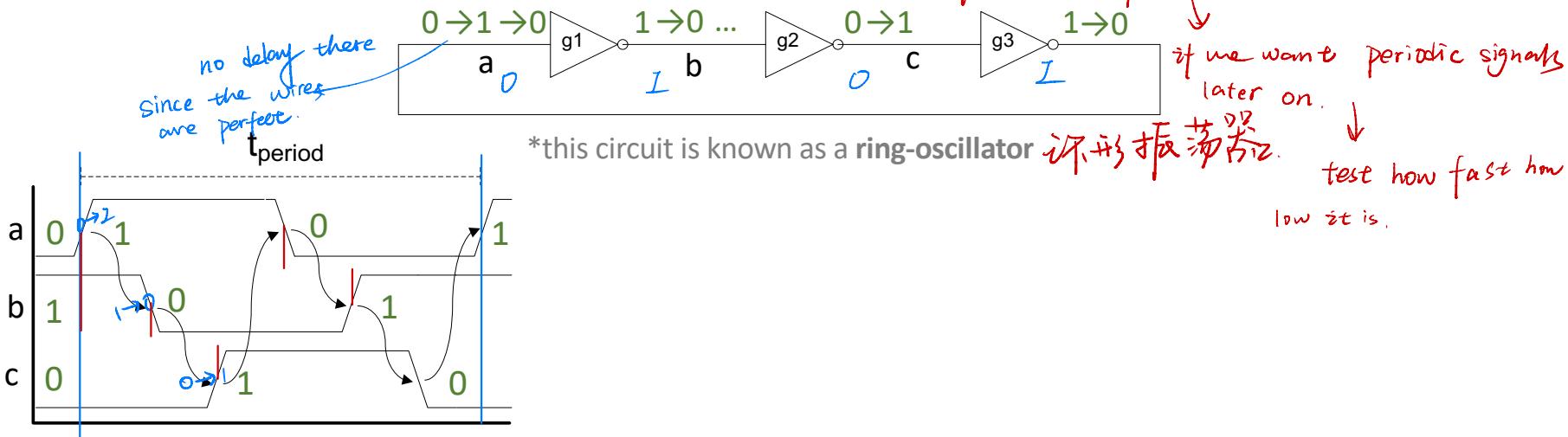
Inputs		Outputs	
a	b	$\bar{a}$	$\bar{a} \cdot \bar{b}$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	0

# Last exercise of the day

## Assumptions

- Low-to-high gate delays are 2 ns ( $t_{pLH}^g = 2 \text{ ns}$ )
- High-to-low gate delays are 1 ns ( $t_{pHL}^g = 1 \text{ ns}$ )
- Wires have zero delay

Task: Calculate the period of the below design

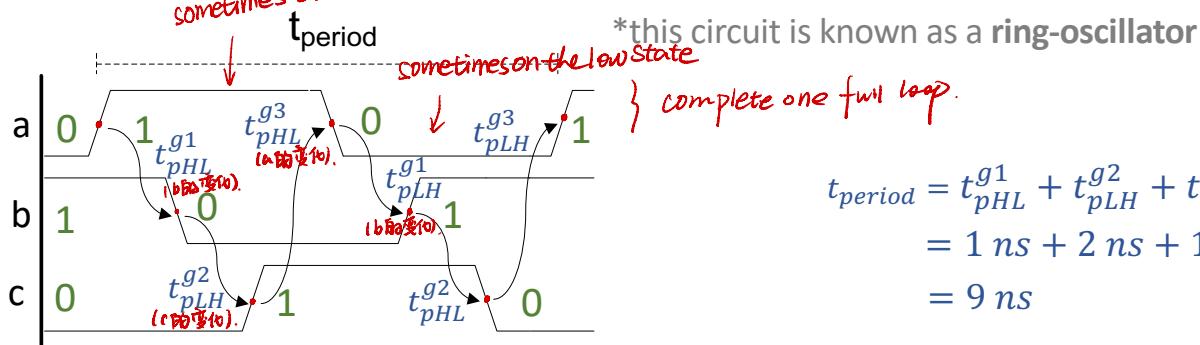
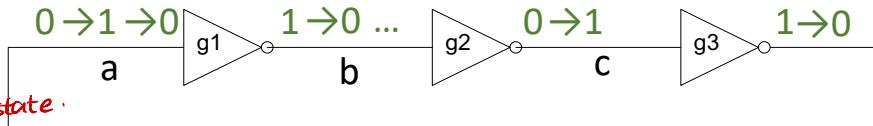


# Last exercise of the day

Assumptions

- Low-to-high gate delays are 2 ns ( $t_{pLH}^g = 2 \text{ ns}$ )
- High-to-low gate delays are 1 ns ( $t_{pHL}^g = 1 \text{ ns}$ )
- Wires have zero delay

Task: Calculate the period of the below design



$$\begin{aligned}t_{\text{period}} &= t_{pHL}^{g1} + t_{pLH}^{g2} + t_{pHL}^{g3} + t_{pLH}^{g1} + t_{pLH}^{g2} + t_{pLH}^{g3} \\&= 1 \text{ ns} + 2 \text{ ns} + 1 \text{ ns} + 2 \text{ ns} + 1 \text{ ns} + 2 \text{ ns} \\&= 9 \text{ ns}\end{aligned}$$

# 1-minute survey



**Link:** <https://docs.google.com/document/d/1PfhK4P3QGfswgygrwLXhP6cWujitIH-4YIkORvkND5w/edit?usp=sharing>

# Questions?

# Questions?